

## **ANALISIS BLOB DETECTION PADA PENDETEKSIAN DAN PERHITUNGAN KENDARAAN DI JALAN TOL**

Deny Nugroho Triwibowo<sup>1</sup>, Ema Utami<sup>2</sup>, Sukoco<sup>3</sup>

Magister Teknik Informatika, Universitas Amikom Yogyakarta<sup>1,2,3</sup>

Email : deny.15@students.amikom.ac.id<sup>1</sup>, ema.u@amikom.ac.id<sup>2</sup>, pak\_koco@yahoo.com<sup>3</sup>

### **ABSTRAK**

Jalan tol dibangun dengan maksud untuk mempercepat dan mengurangi volume lalu lintas pada jalan – jalan utama atau arteri disetiap daerah. Target yang menjadi sasaran pelayanan jasa jalan tol terhadap pemakai jasa adalah kelancaran, keamanan dan kenyamanan. Akan tetapi, pada hari libur ataupun jam sibuk seringkali terjadi penumpukan atau kepadatan volume jumlah kendaraan yang memakai jasa jalan tol, sehingga mengakibatkan terjadinya kemacetan lalu lintas. Hal ini biasanya terjadi akibat kurangnya informasi yang diterima oleh masyarakat akan jumlah kendaraan satu ruas jalan tol. Untuk mengatasi masalah tersebut dibutuhkan sebuah sistem yang dapat mendeteksi dan menghitung jumlah kendaraan yang melewati ruas jalan tol untuk memberikan informasi secara cepat kepada pengendara lain sebelum memasuki jalan tol yang akan dilalui. Metode yang akan dipakai untuk mendeteksi dan menghitung jumlah kendaraan dengan menggunakan *Blob Detection*. Dari hasil penelitian didapatkan nilai error tertinggi sebesar 52% pada video di detik ke-6 sampai detik ke-10. Hal ini disebabkan oleh jumlah kendaraan yang melintas begitu banyak dan dengan kecepatan yang sangat tinggi.

**Kata Kunci:** Jalan Tol, Deteksi Kendaraan, *Blob Detection*.

### **ABSTRACT**

*Toll roads are built with the intention to speed up and reduce the volume of traffic on the main roads or arteries in each area. Targets that become the target of toll road services to service users are fluency, safety and comfort. However, on holidays or rush hours there is often a build-up or volume density of the number of vehicles using toll road services, resulting in traffic congestion. This usually occurs due to lack of information received by the public about the number of vehicles in one toll road section. To overcome this problem, we need a system that can detect and count the number of vehicles passing through toll roads to provide information quickly to other motorists before entering the toll road to be passed. The method will be used to detect and count the number of vehicles using Blob Detection. The results of the study obtained the highest error value of 52% on the video in the 6 second to 10 seconds. This is caused by the number of vehicles passing by so much and at very high speeds.*

**Keywords:** *Blob Detection, Vehicle Detection. Toll Roads*

## 1. PENDAHULUAN

Jalan tol adalah jalan umum yang kepada pemakainya dikenakan kewajiban membayar untuk melewati jalan yang dilalui dan merupakan jalan alternatif lalu lintas dari jalan umum yang telah ada. Sejarah jalan tol di Indonesia dimulai pada tahun 1978 dengan dioperasikannya jalan tol Jagorawi dengan panjang 59 km yang menghubungkan Jakarta, Bogor, dan Ciawi. Pembangunan jalan tol yang dimulai tahun 1975 ini, dilakukan oleh pemerintah dengan dana dari anggaran pemerintah dan pinjaman luar negeri yang diserahkan kepada PT. Jasa Marga (Persero) Tbk. sebagai penyertaan modal. Selanjutnya PT. Jasa Marga ditugasi oleh pemerintah untuk membangun jalan tol dengan tanah yang dibiayai oleh pemerintah (sumber : Badan Pengatur Jalan Tol).

Tingginya tingkat populasi penduduk mulai menjadi masalah khususnya di kota – kota besar. Salah satu akibat yang ditimbulkan adalah kemacetan yang semakin parah. Mobilitas penduduk yang tinggi, bertambahnya jumlah kendaraan setiap tahun dan minimnya pembangunan infrastruktur jalan menambah parahnya kemacetan yang ada. Tidak hanya di jalan raya, jalan tol atau yang biasa disebut jalan bebas hambatan pun sudah mulai mengalami kemacetan. Menurut data yang dikeluarkan PT. Jasa Marga (Persero) Tbk, jumlah volume terpadat terjadi disaat H-7 dan H+3 lebaran tahun 2019 di jalan tol Jakarta - Cikampek. Saat H-7 lebaran tercatat sebanyak 57.405 kendaraan yang melintas di jalan tol tersebut, jumlah itu meningkat sebesar 144 persen dari hari biasanya yang rata – rata mencapai 23.484 kendaraan. Pada saat H+3 lebaran tercatat

ada 166.574 kendaraan yang kembali ke Jakarta. Jumlah ini meningkat sebesar 120 persen dari hari biasanya yang rata – rata mencapai 67.345 kendaraan.

Penelitian yang dilakukan Purwiyanti, dkk (2018), dengan merancang sistem pemantau dan penghitung jumlah kendaraan yang melintas di dua jalur jalan raya dengan metode *blob detection*. Dalam pengujian digunakan 2 video lalu lintas di jalan raya yang berbeda dengan jumlah *frame* sebanyak 900 *frame* pada masing – masing video. Hasil pengujian yang diperoleh dari penelitian yang dilakukan memperlihatkan bahwa sistem ini berhasil menghitung jumlah kendaraan yang melintas di dua jalur pada jalan raya dengan tingkat *error* rata – rata sekitar 10.3 %.

Penelitian yang dilakukan Meru & Mujawar (2015) mengusulkan sebuah sistem untuk mendeteksi dan mengontrol lalu lintas, di mana sistem yang akan dibuat menggunakan metode *background subtraction*. Dari hasil percobaan, ditunjukkan bahwa metode *background subtraction* yang diterapkan untuk mendeteksi kendaraan lebih baik dan akurat. Evaluasi dilakukan untuk segmentasi yang lebih baik dari kendaraan yang bergerak dan menghasilkan hasil yang lebih baik dalam *microsecond*.

Penelitian berikutnya yang dilakukan Hidayati (2017), dengan menerapkan metode *blob detection* untuk memperkirakan dalam deteksi kendaraan. Pengujian yang dilakukan menggunakan video dengan durasi 5 menit, *frame rate* sebesar 30 *fps*, dan video yang diinputkan berukuran 480 x 320 piksel. Didapatkan tingkat akurasi pengujian sistem sebesar 82,11% untuk mendeteksi kendaraan pada kondisi yang cerah dan pada saat kondisi hujan didapatkan nilai akurasi

sebesar 76,50%. Untuk arus lalu lintas dalam keadaan lengang didapatkan nilai 83,07%, sangat lebih baik ketimbang arus pada kondisi ramai yang hanya mendapatkan nilai sebesar 67.70%. Hal ini disebabkan karena posisinya yang bergerombol dan berdekatan satu sama lain.

Kautsar & Adi (2016), dalam penelitiannya untuk mengembangkan sistem penghitung dan deteksi kendaraan otomatis yang akurat sebagai pemantau kontrol lalu lintas dan analisa lalu lintas dengan metode Kalman Filter dan Gaussian Mixture Model (GMM). Dalam pengujian peneliti telah mengakuisisi data yang dilakukan pada 4 variasi yaitu, waktu pada pagi hari, siang hari, sore hari, dan malam hari untuk mengetahui pengaruh pencahayaan pada hasil processing. Hasil pengujian yang dilakukan menunjukkan bahwa implementasi metode ini dalam mendeteksi dan menghitung kendaraan mendapatkan hasil yang paling akurat pada pagi hari, sedangkan penghitungan kendaraan yang paling tidak akurat terjadi pada malam hari. Hal ini disebabkan oleh intensitas pencahayaan berkurang pada malam hari.

Berdasarkan dari latar belakang masalah dan beberapa penelitian terkait sebelumnya, tujuan studi ini adalah menganalisis metode *blob detection* dalam melakukan deteksi jenis kendaraan serta menghitung kendaraan yang terdeteksi. Terdapat beberapa batasan masalah dalam Studi ini. Batasan masalah ini meliputi, resolusi kamera yang digunakan, sudut pandang objek kendaraan terhadap kamera, format file video, kondisi cuaca, serta nilai fps video yang digunakan. File video yang digunakan berformat MP4. Sudut pandang objek kendaraan terhadap kamera adalah

tampak depan dari tengah jalan. Kondisi cuaca cerah dan siang hari.

## 2. LANDASAN TEORI

### 2.1 Citra Digital

Citra digital merupakan representasi dari citra yang diambil oleh mesin dalam bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom (*pixel*) dan kuantisasi menyatakan besar kecilnya tingkat kecerahan (*grayscale*) sesuai dengan jumlah *bit biner* mesin yang digunakan untuk mendapatkan citra tersebut (Basuki, Palandi, & Fatchurrochman, 2005).

Menurut Munir (2004), Citra dibagi menjadi dua jenis yaitu :

- a. Citra diam (*still images*), citra tunggal yang tidak bergerak.
- b. Citra bergerak (*moving images*), merupakan rangkaian citra diam yang ditampilkan secara beruntun (*sequential*) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Contohnya adalah gambar-gambar yang terlihat pada televisi atau layar lebar.

### 2.2 Blob Detection

*Blob detection* adalah mendeteksi kumpulan pixel-pixel yang memiliki warna yang sama dibandingkan dengan latar belakangnya agar dapat mendeteksi *low-level* dalam suatu objek baik dua dimensi maupun tiga dimensi. *Blob* dapat ditampilkan dengan acara yang berbeda-beda tergantung metode yang akan digunakan karena setiap metode memiliki kelebihan dan kekurangannya masing-masing. Berikut ini beberapa metode yang

sering digunakan dalam *blob detection* yaitu (Kaspers, 2009).

- a. *Prerequisites* adalah berdasarkan representasi skala-ruang. Utama tujuan dari representasi ruang-skala adalah untuk memahami struktur gambar di semua level resolusi secara bersamaan dan tautkan gambar di skala berturut-turut.
- b. *Template Matching* adalah mendeteksi templet yang cocok dari bagian *sample* gambar.
- c. *Watershed Detection* adalah mengasumsikan sebuah gambar menjadi pegunungan dengan nilai abu-abu dan mensimulasikan proses hujan yang jatuh ke pegunungan, berlari menuruni pegunungan dan terakumulasi dalam cekungan.
- d. *Spoke Filter* adalah yang mendeteksi gumpalan berbagai ukuran adalah *Spoke Filter* juga disebut *Adaptive Spatial Erosion Filte* yang diusulkan oleh Minor & Sklansky (1981).
- e. *Automatic scale selection* adalah Sebagian besar aplikasi deteksi gumpalan didasarkan pada metode Lindeberg untuk pemilihan skala otomatis.
- f. *Effective maxima line detection* adalah menyajikan metode di mana kurva terhubung modulus maxima pada skala yang berbeda - disebut garis maxima - dipilih secara efektif, untuk membagi gumpalan dari kebisingan.
- g. *Confidence Measurement* adalah menyajikan metode multi-skala yang agak rumit untuk mengekstrak gumpalan dari

sebuah gambar. Ini tidak didasarkan pada *smoothing Gaussian linear*, seperti metode sebelumnya.

- h. *Sub-pixel precise blob detection* terdiri dari dua langkah utama dalam metode sub-pixel precise *blob detection*. Langkah pertama terdiri dari ekstraksi titik pusat potensial gumpalan dalam presisi subpiksel. Langkah kedua diuraikan dalam terdiri dari merekonstruksi batas di sekitar titik tertentu (Hinz, 2005).

Ekstraksi titik pusat *blob detection* didasarkan pada geometrik diferensial, persegi panjang dari titik pusat dimana panjang  $l$ , dan lebar  $w=(w<l)$  dan daerah homogen, yaitu kontras yang konstan  $h$ , berfungsi sebagai primitif dasar. Dibutuhkan algoritma *smoothing* agar dapat menekan gangguan. Oleh karena itu, tujuan dari algoritma adalah pertama untuk menemukan titik-titik pusat dan kemudian merekonstruksi batas persegi panjang. Mengasumsikan bahwa persegi panjang dapat berorientasi sepanjang sumbu koordinat  $x$ ., Kemudian fungsi model  $fr$  untuk persegi panjang dengan kontras yang dinormalisasi  $h=1$  diberikan oleh:

$$fr(x,y) = 1 \text{ dari } |x| \leq d \text{ dan } |y| \leq w$$
$$fr(x,y) = 0 \text{ maka} \quad (1)$$

dan respon  $r\sigma$  setelah mengubah  $fr$  dengan kernel *Gaussian smoothing* satu dimensi

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

dan pada *Gaussian smoothing* dua dimensi

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

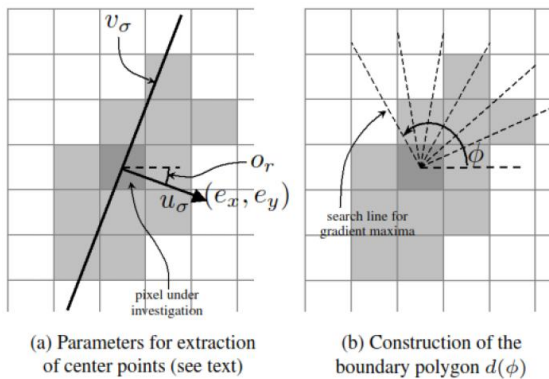
sehingga menjadi

$$r_{\sigma}(x, y, l, w) = (g | |\sigma(x + 1) - g_{\sigma}(x - 1)|)(g_{\sigma}(y + w) - g_{\sigma}(y - w)) \quad (4)$$

orientasinya dapat ditentukan dengan menghitung *eigenvector* dari *Hessian Matrix* H

$$H(x, y) = \begin{vmatrix} r_{xx} & r_{xy} \\ r_{yx} & r_{yy} \end{vmatrix} \quad (5)$$

dan memilih *eigenvector* ( $e_x; e_y$ ) yang sesuai dengan nilai eigen absolut yang lebih besar  $\lambda_e$  ( $\lambda_e < 0$  untuk gumpalan terang  $\lambda_e > 0$  untuk gumpalan gelap).



Gambar 1 Menentukan titik pusat pada *Sub-pixel precise blob detection*

### 2.3 Computer Vision

Computer vision adalah cabang ilmu komputer dan rekayasa yang memiliki tujuan untuk membuat komputer yang dapat melihat dan mengerti kejadian di dunia luar (Ikeuchi, 2014 ). Computer vision dikhususkan untuk menemukan algoritma, representasi data, dan arsitektur komputer yang mewujudkan prinsip-prinsip yang mendasari kemampuan visual (Liu et al., 2009). Computer vision merupakan kombinasi antara pengolahan citra dan pengenalan pola. Berikut adalah bagian dari computer vision :

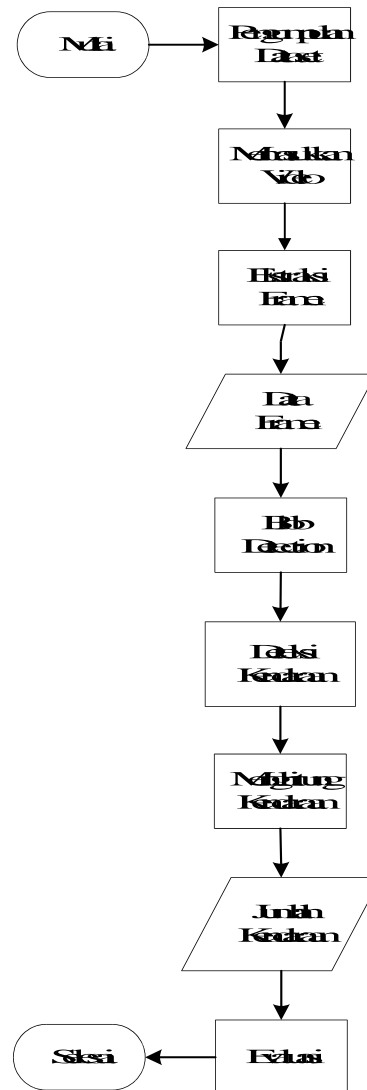
- i. Pengolahan Citra : bidang yang berhubungan dengan proses transformasi citra/gambar. Proses

ini bertujuan untuk mendapatkan kualitas citra yang lebih baik.

- j. Pengenalan Pola : bidang yang berhubungan dengan proses identifikasi objek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi/pesan yang disampaikan oleh gambar/citra.

### 3. METODE PENELITIAN

Adapun metode penelitian dapat dilihat pada Gambar 2.



Gambar 2 Alur Penelitian

### 3.1 Pengumpulan Data

Dataset yang digunakan terdiri dari 100 citra training (75 citra positif berupa citra mobil dan 25 citra negatif berupa citra bukan mobil) dan sudut pandang citra mobil ialah bagian depan mobil. Pengujian dilakukan dengan menggunakan data berupa video yang berisi kondisi jalan dengan spesifikasi sebagai berikut.

- a. Durasi video 10 detik.
- b. Frame rate sebesar 20 fps.
- c. Ukuran video 480 x 320 piksel.

### 3.2 Input Video

Menginputkan file video ke dalam program yang di ambil dari kamera pemantau (CCTV) yang terdapat di jalan toll yang akan digunakan sebagai bahan utama untuk proses analisa.

### 3.3 Ekstraksi Frame

Objek kendaraan yang terdapat pada video yang diinputkan ke dalam program tersebut kemudian diekstraksi ciri- cirinya, sehingga tiap-tiap objek tersebut akan muncul ciri-cirinya masing-masing, ini adalah tahap dimana citra kendaraan akan dibandingkan dengan background-nya. Dengan mendapatkan kesimpulan jika objek kendaraan yang melewati background akan muncul nilai.

### 3.4 Data Frame

Setelah dilakukan ekstraksi ciri pada objek kendaraan, maka akan keluar nilai yang memuat intensitas piksel yang terdapat di dalam tiap framenya. Nilai ini akan menjadi acuan data yang dapat menentukan ciri – ciri dari objek kendaraan tersebut.

### 3.5 Blob Detection

Mendeteksi nilai atau gumpalan dalam objek kendaraan yang berbeda dalam ciri - cirinya seperti kecerahan atau warna yang dibandingkan dengan backgroundnya.

### 3.6 Deteksi Kendaraan

Memisahkan tiap- tiap gumpalan yang terdapat dalam suatu objek dan kemudian melabelinya sehingga antara gumpalan yang satu dengan gumpalan yang lainnya dapat dibedakan dengan ciri - ciri yang berbeda.

### 3.7 Jumlah Kendaraan

Proses perhitungan jumlah kendaraan dilakukan dengan cara menghitung jumlah label yang terdeteksi pada objek kendaraan yang melintas.

### 3.8 Evaluasi Sistem

Proses terakhir adalah melakukan evaluasi terhadap system atau program yang telah dibuat dengan menghitung nilai error yang terjadi pada saat perhitungan jumlah kendaraan dengan system dan perhitungan jumlah kendaraan dengan manual.

## 4. HASIL DAN PEMBAHASAN

Setelah pembuatan program selesai dilakukan, contoh hasil eksekusi sistem dapat dilihat pada Gambar 2. Dengan menginputkan video yang telah diekstraksi menjadi beberapa *frame*.



Gambar 3 Frame Asli

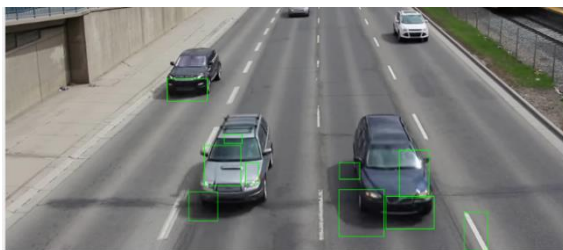
Pada Gambar 3 adalah hasil dari *blob detection*, dimana obyek terdeteksi dan dipisahkan dari *background*-nya. Setelah itu dilakukan *blob tracking*, yaitu

memberikan tanda kotak pada obyek hasil *blob detection*.



Gambar 4 Blob Deteksi

Selanjutnya yang ditunjukkan pada Gambar 4. hasil dari blob detection dihitung berapa banyak kotak yang melintasi garis deteksi, dan hasil perhitungan lalu ditampilkan di layar komputer.



Gambar 5 Deteksi Kendaraan

Pengujian lalu dilakukan untuk mengetahui performa dari sistem yang dihasilkan, kemudian hasilnya dibandingkan dengan hasil penghitungan secara manual. Tingkat kesalahan (*error*) lalu dihitung menggunakan rumus (Hidayati, 2017) :

$$\%error = \frac{\text{perhitungansistem} - \text{manual}}{\text{perhitunganmanual}} * 100\% \quad (1)$$

Tabel 1 Hasil Pengujian

Waktu (Detik)	Jumlah Kendaraan		Error %
	Sistem	Manual	
00 - 03	6	6	0
03 - 06	21	15	40%
06 - 10	38	25	52%

Tabel 1 memperlihatkan hasil pengujian untuk mendeteksi jumlah kendaraan, dengan jumlah waktu sebanyak 10 detik. Dapat terlihat bahwa persentase *error* sebesar 0 % untuk detik ke-0 sampai detik ke-3 atau dengan kata lain sistem telah melakukan pendeteksian dengan benar karena jumlah kendaraan yang terlintas masih tidak terlalu banyak. Namun untuk detik ke -3 sampai detik ke-6 mulai terlihat perbedaan perhitungan dengan jumlah *error* 40%. Hal ini mungkin disebabkan faktor intensitas jumlah kendaraan bertambah dan kecepatan kendaraan juga bertambah. Selanjutnya untuk detik ke-6 sampai detik ke-10 terlihat jumlah perbandingan yang signifikan dengan jumlah *error* sebesar 52%. Hal ini disebabkan karena runtutan kendaraan yang melintas saling beriringan dan kecepatan kendaraan juga begitu sangat cepat. Artinya satu obyek dapat terhitung lebih daripada satu saat nilai jumlah kendaraan yang melintas begitu banyak dan dipacu dengan kecepatan tinggi.

## 5. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

- Deteksi kendaraan menggunakan *blob detection* didapatkan nilai *error* tertinggi sebesar 52%.
- Nilai *error* didapatkan dari kendaraan yang melesat sangat cepat memberikan masalah pada *counting*, meskipun kendaraan berhasil di-*tracking*.
- Arus lalu lintas yang cukup padat dengan kecepatan rendah menghasilkan

bertumpuknya kendaraan dan berhimpitan karena adanya bayangan dari kendaraan, sehingga kendaraan yang berhimpit tersebut dikenali sebagai satu kendaraan yang utuh.

## 5.2 Saran

Untuk mengatasi permasalahan tersebut untuk pengembangan selanjutnya diperlukan metode blob tracking yang lebih baik. Walaupun demikian, hasil yang didapat merupakan studi awal yang baik bagi mengembangkan teknologi di bidang transportasi.

## DAFTAR PUSTAKA

- Badan Pengatur Jalan Tol, Sejarah Jalan Tol, diakses pada 13 Januari, 2019, dari <http://bpjt.pu.go.id/konten/jalan-tol/sejarah>
- Basuki, A., Palandi, J., & Fatchurrochman. (2005). Pengolahan citra digital menggunakan Visual Basic. Yogyakarta: Graha Ilmu.
- Hidayati, Q. (2017). Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection. *JNTETI*, 6(2), 215–221.
- Hinz, S. (2005). Fast And Subpixel Precise Blob Detection and Attribution. *Ieee International Conference on Image Processing*, 3, 457. Retrieved from [http://pdf.aminer.org/000/323/513/fast\\_and\\_subpixel\\_precise\\_blob\\_detection\\_and\\_attribution.pdf](http://pdf.aminer.org/000/323/513/fast_and_subpixel_precise_blob_detection_and_attribution.pdf)
- Ikeuchi, K. (2014 ). Computer Vision : A Reference Guide. Springer.
- Kaspers, A. (2009). Blob Detection Anne. *Biomedical Image Sciences Image Sciences Institute*. Belanda: UMC Utrecht.
- Kautsar, H. V. Al, & Adi, K. (2016). Implementasi Object Tracking Untuk Mendeteksi dan Menggunakan Metode Kalman Filter dan Gaussian Mixture Model. *Youngster Physics Journal*, 5(1), 13–19.
- Liu, Q., Li, X., Elgammal, A., Hua, X. sheng, Xu, D., & Tao, D. (2009). Introduction to computer vision and image understanding the special issue on video analysis. *Computer Vision and Image Understanding*, 113(3), 317–318.  
<https://doi.org/10.1016/j.cviu.2008.11.004>
- Meru, A. V, & Mujawar. (2015). Computer Vision Based Vehicles Detection and Traffic Control For Four Way Road. *Tecnical Research Organisation India*, 2(6), 20–25.
- Minor, L. G., & Sklansky, J. (1981). The Detection and Segmentation of Blobs in Infrared Images. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(3), 194–201.
- Munir, R. (2004). Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.
- Purwiyanti, S., Setyawan, F. X. A., Hafizulahudin, M., Herlinawati, & Murdika, U. (2018). Sistem Penghitung Jumlah Kendaraan Di Lintasan Dua Arah Menggunakan Library CVBlob. *Seminar Nasional Teknik Elektro*, 296–299.